Interleaved Challenge Loop PUF: A Highly Side-Channel Protected Oscillator-Based PUF

Lars Tebelmann, Jean-Luc Danger, Michael Pehl

Interleaved Challenge Loop PUF: A Highly Side-Channel Protected Oscillator-Based PUF

Lars Tebelmann, Jean-Luc Danger *Member, IEEE*, Michael Pehl *Member, IEEE*

***Abstract*—Physical Unclonable Functions (PUFs) leverage manufacturing variations to generate device-specific keys during runtime only, overcoming the need for protection after power-off as for Non-Volatile Memory. The main challenges of PUF-based key storage are reliability of the response and sensitivity to Side-Channel Analysis (SCA). Oscillator-based PUFs are particularly sensitive to frequency spectrum SCA. Existing countermeasures can protect sign-based bit derivation that requires error correction or discarding unreliable bits to achieve reliable key generation. Amplitude-based bit derivation enhances the reliability of oscillator-based PUFs without discarding unsteady response bits, keeping a high entropy. However, existing lightweight countermeasures are not applicable for this case. This raises the demand for an alternative solution. This work targets the protection of amplitude-based bit derivation combined with the Loop PUF, an oscillator-based PUF primitive well suited for key generation. It presents the Interleaved Challenge Loop PUF (ICLooPUF), a side-channel-hardened offspring of the Loop PUF that uses dynamic challenge interleaving. The SCA-protected PUF primitive is applicable to amplitude-based and sign-based bit derivation methods, and requires a low hardware overhead. Theoretical and experimental results show the efficiency and effectiveness of the protection mechanism.**

***Index Terms*—Physical Unclonable Function, Side-Channel Analysis, Loop PUF, Countermeasure, Two-Metric Helper Data, Equiprobable Quantization, Lehmer-Gray Order Encoding.**

I. Introduction

PUFs provide means for secure key storage on embedded low-cost devices that do not require expensive secured Non-Volatile Memory (NVM). By leveraging manufacturing variations for each device a unique, but reproducible PUF response is generated that can be used to embed a secret key for further cryptographic operations. Compared to classical key storage, the secret is only derived on-demand from the PUF and stored in volatile memory, i.e., during power-off no key material remains on the device, which reduces the attack surface. Besides criteria that evaluate the PUF's quality such as uniqueness, reliability and unpredictability, protecting PUFs against SCA is a major concern for practical applications.

The Interleaved Challenge Loop PUF (ICLooPUF) introduced in this work is an enhancement of the Loop PUF [1], an oscillator-based PUF configured by challenges. When used as a cryptographic key generator, a fixed set of challenges like Hadamard codes allow to get the required key entropy. We take the Loop PUF as the origin of our research, since it is an easy-to-design, area-efficient PUF providing high quality response bits, which makes it, as well as our new design, a good candidate for key generation.

However, previous work demonstrated the feasibility of SCA for oscillation-based PUF primitives such as the Ring-Oscillator (RO) [2]–[4], Transient Effect Ring-Oscillator (TERO) [5], [6] and Loop PUFs [7], based on the fact that frequency-related emanations are observable by an attacker. As usually the sign of a difference of two frequency measurements is used as a secret bit, reconstruction of the secret from this side-channel is possible by comparing observations. An effective protection scheme for oscillation-based PUFs is the *temporal masking* countermeasure [7] that randomizes the order of observations and thus impedes SCA attacks by protecting the sign of the frequency difference. The required randomness is derived from jitter of the oscillator forming the PUF and does not require an additional True Random Number Generator (TRNG) circuit, which guarantees a low implementation overhead.

A major drawback of the sign-based bit derivation is that environmental and dynamic noise leads to low reliability for bits derived from frequency differences close to zero. Possible solutions to increase reliability include discarding unreliable bits, at the cost of decreasing the entropy of the PUF response, or requiring stronger error-correction, at the cost of increased time and hardware overhead. Applying these solutions also introduces additional attack vectors.

An alternative is to better exploit the randomness from the manufacturing process, for instance by using amplitude-based bit derivation schemes. These schemes exploit not only the sign of the difference of two oscillator observations, but also their absolute value. In combination with the Loop PUF, the Two-Metric Helper Data (TMHD) scheme [8] has been suggested from this category. Compared to sign-based bit derivation, it allows for enhancing the reliability by changing the threshold of the response extractor to a non-zero amplitude value. Additional helper data in this scheme allow for reliably restoring all response bits from noisy measurements without discarding the unsteady bits.

However, the TMHD is not protected against SCA by the *temporal masking* countermeasure: As soon as the absolute value of a frequency difference is used in the secret bit derivation, randomization of the sign provides no protection [9]. A

Lars Tebelmann and Michael Pehl are with the Chair of Security in Information Technology, TUM School of Computation, Information and Technology, Technical University of Munich.

Jean-Luc Danger is with Télécom Paris, Institut polytechnique de Paris.

This work was partly funded by the Agence Nationale de la Recherche (ANR) under grant number ANR-20-CYAL-0007 and the German Federal Ministry of Education and Research (BMBF) under grant number 16KIS1389K within the project APRIORI.

countermeasure suggested for the TMHD is based on further randomizing the order of frequency comparisons. But this requires a substantial amount of random bits, does not have a guaranteed maximum runtime, and exhibits a significant complexity increase [9].

Contributions: In order to provide robustness against SCA with low overhead for amplitude-based bit derivation schemes like TMHD, we propose the dynamic change of challenges, termed as *challenge interleaving*, for the Loop PUF resulting in the ICLooPUF. The proposed technique thwarts SCA by breaking the relation between spectral emanations of the implementation and the frequency difference of the challenges, i.e., the secret response. In particular, our contributions comprises:

- Proposal of a novel architecture called ICLooPUF, which derives from the original Loop PUF.
- Proof-of-concept implementation of the ICLooPUF on FPGA showing the practical feasibility.
- Theoretical validation of the resistance of the ICLooPUF against SCA.
- Practical evidence that the ICLooPUF does indeed not reveal exploitable side-channel leakage.

Structure: We first provide required background regarding the Loop PUF in Section II and introduce bit derivation schemes for which the new protection mechanism is beneficial. In Section III, we define the attacker model, and introduce and discuss the ICLooPUF in Section IV. In Section V we present our proof-of-concept design and demonstrate practical evaluation results. After a discussion in Section VI, we conclude our work in Section VII.

II. Background

The Interleaved Challenge Loop PUF (ICLooPUF) is an enhancement of the Loop PUF. The goal of the construction is to avoid side-channel leakage independent of the bit derivation method. As a background, this section first recapitulates the basic concept of the Loop PUF. Subsequently, we introduce sign-based and amplitude-based bit derivation methods and discuss possible SCA attack vectors that have to be mitigated by a countermeasure.

A. The Loop PUF

The Loop PUF [1] is an oscillation-based PUF primitive consisting of a chain of N delay elements as depicted in Fig. 1. All delay elements $j \in \{1, \ldots, N\}$ are designed to be nominally the same and consist of two possible delay paths as shown in Fig. 2. The delay used to forward the input to the output is set by a multiplexer controlled by the challenge bit c_j.

The last delay element is fed back through an inverter to the input of the Loop PUF. Due to the odd number of inverters in the system, the Loop PUF starts oscillating as soon as it is enabled, e.g., by putting an AND gate with an additional enable input into the feedback. A reference counter (not depicted in Fig. 1) counts the cycles of a sample clock with period T_{ref}, and stops the oscillation as soon as it reaches a predefined value n_{max}. The oscillation frequency f_{loop} of the Loop PUF depends on the overall delay of the structure in Fig. 1 determined by the challenge $C = [c_1, \ldots, c_N]$. A counter at the output counts the number of oscillations for the time defined by n_{max}, resulting in a counter value of

$$n_{loop} = n_{max} \cdot T_{ref} \cdot f_{loop}. \quad (1)$$

The bits k_i in the Loop PUF design are derived from the difference $\Delta_i = n_{loop,i} - n_{loop,\neg i}$ between the counter values $n_{loop,i}$ and $n_{loop,\neg i}$ for a challenge C_i and its inverse $\neg C_i$. This way a good reliability of the Loop PUF is achieved.

The challenges of the Loop PUF are defined to be Hadamard codewords of length N that have a Hamming weight $\frac{N}{2}$ [10]. The all-zero and all-one Hadamard codewords are excluded from the challenge space, so that challenges C_i with $i \in \{1, \ldots, N-1\}$ are used and $N-1$ bits are derived from a Loop PUF with N stages. Since all delay elements are identically designed, applying only challenges of this form ensures that the same number of upper and lower paths is selected and bias introduced through imbalance of the upper and lower path in Fig. 2 is eliminated. Further, since Hadamard codewords mutually have Hamming distance $\frac{N}{2}$, between all challenges half of the delay elements differ, which maximizes the entropy.

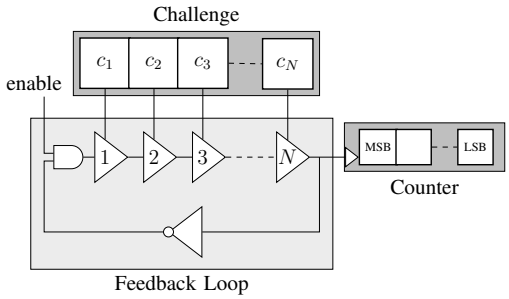


Fig. 1. Schematic of the Loop PUF (modified from [7]).

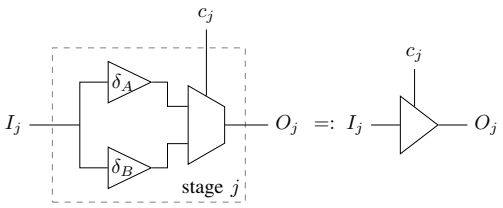


Fig. 2. Internal structure of a delay element (modified from [7]).

B. Sign-Based Bit Derivation

The Loop PUF's counter difference Δ_i has to be mapped to a response bit k_i. In the original proposal [1], the sign of the difference is used, e.g., $k_i = 0$ for $\Delta_i \geq 0$ and $k_i = 1$ otherwise. Regarding SCA, an attacker observing the frequency f_{loop} for both challenges independently retrieves the sign of their difference and thus the respective bit. The

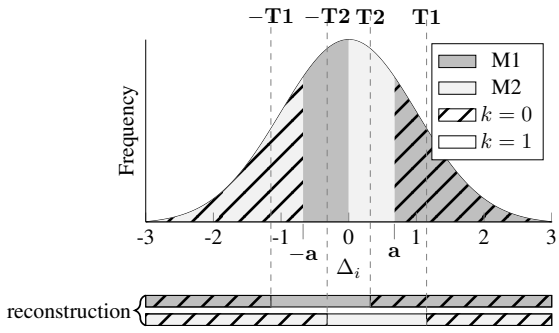


Fig. 3. Two-Metric Helper Data scheme according to [8] (modified from [9]).

attack is possible under the assumption that the order of C_i and $\neg C_i$ is known to the attacker. The *temporal masking* countermeasure [7] randomizes the measurement order of C_i and $\neg C_i$ and thwarts SCA attacks. The required random bit is derived from the jitter of the Loop PUF oscillation, i.e., there is no need for a Random Number Generator (RNG) making *temporal masking* a lightweight countermeasure.

The main drawback of the sign-based approach is that it provides bits that require a strong error correction. The reason is that noise and environmental changes may change the sign of Δ_i if is close to 0 causing faulty bits, i.e., bits, for which the response deviates from the expected/enrolled value.[1] A simple method to improve reliability is to drop unreliable bits, a method known as *dark-bit masking*. This, however, wastes PUF bits – i.e., reduces the entropy – and introduces an attack vector through helper data manipulation detailed in Section III.

C. Amplitude-Based Bit Derivation Methods

In this section, we outline schemes that derive PUF bits based on the amplitude of Δ_i and improve the reliability or the number of extracted bits compared to the sign-based method. In addition, we show the impact of side-channel attacks revealing information about the amplitude of a frequency difference.

1) The Two-Metric Helper Data Scheme: The TMHD [8] visualized in Fig. 3 subdivides the assumed normal distribution of the counter differences into octiles. During an enrollment phase, the difference Δ_i of the Loop PUF under C_i and $\neg C_i$ is measured. Depending on the quantile Δ_i falls into, the response bit k_i is selected and the metric is stored, under which the response bit is most robust against noise. When the PUF response is reconstructed later to derive the same key as in the enrollment phase, the distribution of the counter differences is measured and the octiles are recomputed. The bits are derived based on the stored metric. If for instance a counter difference falls into the second quantile ($-a < \Delta_i < 0$) during enrollment, the metric M1 is stored and the derived secret bit is $k_i = 1$ according to Fig. 3. If during reconstruction Δ_i shifts due to noise, e.g., into the second octile ($-T1 < \Delta_i \leq -a$), still decoding under metric $M1$ would result in $k_i = 1$. The described principal makes the TMHD scheme very robust against noise.

Regarding SCA analysis, the TMHD scheme decodes a counter difference $|\Delta_i| < a$ to $k_i = 0$ and $|\Delta_i| > a$ to $k_i = 1$. Therefore, an attacker who can observe $|\Delta_i|$ can derive the secret bit. Note that *temporal masking* only protects the sign and does not hinder attacks on the TMHD. Further, modifying the mapping of the secret bit in Fig. 3, e.g., such that $k_i = 0$ for $\Delta_i \geq 0$ does not hinder SCA attacks either as shown in [9]: the helper data of the TMHD is publicly known and an attacker can combine it with SCA observations to retrieve the key. Therefore, a countermeasure must protect the amplitude of the frequency difference $|\Delta_i|$ from observation by an attacker.

2) Equiprobable Quantization: Equiprobable Quantization (EQP) has been introduced in the context of PUFs as zero leakage quantization [14] and for tamper-evident PUFs [15]. Similarly to the TMHD scheme, the distribution of analog values – in the case of the Loop PUF counter differences – is divided into intervals of equal probability. Each interval is assigned a symbol from a higher order alphabet. Compared to the TMHD scheme typically more bits of entropy are derived at the cost of lower reliability. Without leaking secret information, robustness is gained by relating the minimum size of an interval to the expected noise level and storing helper data describing the position in the interval [14].

Again, the quantization method has to be considered publicly known, i.e., an attacker measuring Δ_i including the sign can compute the quantization, and the symbol derived on the device. Even if the attacker can only reveal $|\Delta_i|$ without the sign, only two intervals are possible, one for $|\Delta_i|$ and one for $-|\Delta_i|$. So although protection of the sign is achieved, e.g., by a temporal masking scheme [7], the remaining entropy for an attacker would be reduced to one bit rendering the approach useless compared to sign-based bit derivation.[2] Therefore, EQP is only useful if an attacker cannot learn about the amplitude of the counter difference $|\Delta_i|$.

3) Order encoding: Lehmer-Gray order encoding for PUFs has been suggested to derive a large number of stable bits from an array of RO PUFs [16]. The RO frequencies, respectively counter values measuring it, are sorted with respect to their value. Each possible order is assigned a unique bit sequence used as response. The sequence is encoded under the constraint of a low sorting overhead and such that minimal changes in the ordering leads to a small variation in the derived bit sequence. In order to remove bias effects for specific oscillator positions, the authors in [16] subtract for each position an individual offset from the counter.

Using Hadamard challenges, bias is largely removed and order encoding could be directly applied to the counter differences Δ_i of a specific Loop PUF instance without subtracting an offset. However, the same problem as for the other amplitude-based methods appears: An attacker observing all possible Δ_i or at least $|\Delta_i|$ can make strong statements about

[1] For PUF key generation, the key is commonly fixed during an enrollment process and is reconstructed later from noisy PUF measurements using a helper data algorithm and error correction [11]–[13].

[2] The argumentation also holds for any other oscillator-based PUF, for which an attacker observes the analog properties. Furthermore, it applies to other quantization schemes like equidistant quantization.

the resulting order. Therefore, not only the sign but also the amplitude of Δ_i has to be protected.[3]

D. Protection of the Loop PUF

Table I summarizes SCA attack vectors on the Loop PUF for the different bit derivation methods and compares existing countermeasures. For the sign-based method, *temporal masking* (TM) [7] provides a low-cost countermeasure that efficiently hides the sign of the counter difference from an attacker. However, if applied to amplitude-based schemes, the attacker is able to significantly reduce entropy or to completely break the system, if $|\Delta_i|$ or Δ_i can be measured.

In order to protect the Loop PUF independently of the bit derivation method, either the order of challenges $C_1, \ldots, C_N$ can be randomized, i.e., an attacker does not know the correct order of her observations, or $|\Delta_i|$ must be hidden.

The first approach, referred to as *challenge randomization* (CR) [9], randomizes the challenge index i: While the device can resolve the correct order, an attacker is not able to sort SCA observations accordingly. However, this approach requires a substantial number of random bits from a TRNG to achieve protection [9] and is too complex for lightweight applications. Furthermore, the number of required randomness as well as the runtime for challenge randomization can only be estimated on average and is not deterministic.

In the following, we opt for the second approach and propose *challenge interleaving* as a low-complexity modification of the Loop PUF that hides side-channel leakage without the need of additional randomness and is applicable to amplitude- and sign-based bit derivation.

TABLE I
COMPARISON OF BIT DERIVATION METHODS REGARDING SIDE-CHANNEL ATTACK VECTOR AND COUNTERMEASURE PROTECTION. TM=TEMPORAL MASKING, CR=CHALLENGE RANDOMIZATION.

	Attack Vector		Protection level		
	Sign	Magn.	TM [7]	CR [9]	this work
Sign-based [1]	x		full	full	full
TMHD [8]		x	none	full	full
EPQ [15]	x	x	sign only	full	full
Order Enc. [16]	x	x	sign only	full	full

III. ATTACKER MODEL

In this work, we consider an attacker with physical access to the device. The attacker is able to measure side-channel information related to the PUF, e.g., from power leakage, electromagnetic emission, or light emanations. We consider only attacks on the PUF primitive; attacks on further processing like subsequent error correction or hashing (e.g., [17]–[20]) are out of scope. We consider the scenario of PUF-based key storage, i.e., all helper data – including the one used for bit derivation – have to be considered public and reading them out is in the scope of an attacker. Furthermore, we assume that implementation details, such as the order of challenges $C_1, \ldots, C_N$ that are applied to the Loop PUF and the measurement duration defined by n_{max} and T_{ref}, are known by the attacker. Note that in the scenario of key generation with the Loop PUF or the ICLooPUF, machine learning attacks like in [21] are not applicable, because a very limited number of challenges is applied to a single Loop PUF.

The primary attack vector for oscillator-based PUFs is the frequency spectrum. We consider observing single periods of the oscillation in the time domain practically infeasible due to noise and required measurement precision. However, an attacker measuring the spectrum of a Loop PUF under challenges C_i and $\neg C_i$ can relate the observed frequencies to counter values and derive the sign and the amplitude of the counter difference Δ_i. Therefore, without countermeasures, sign-based and amplitude-based bit derivation schemes are prone to SCA as summarized in Section II.

While attacks on the sign are prevented by using *temporal masking* [7], in order to exploit manufacturing-caused randomness more efficiently with the PUF, amplitude-based quantization algorithms like from Section II-C are required. In this work we focus on the TMHD, as a very efficient method to increase reliability without requiring an error correction decoder; As a consequence, TMHD does not waste response bits to store redundancy and has very low overhead regarding helper data. Our findings regarding SCA apply, however, to all other amplitude-based schemes, too.

Since an attacker might combine attack vectors, another commonly known weakness of some quantization schemes, which has to be mentioned, is the vulnerability to Helper Data Manipulation Attacks (HDMA). HDMA have been shown for schemes that select stable PUF bits in order to increase reliability [22], [23], and also for sign-based bit derivation [8]. However, the TMHD scheme does not suffer from such attacks: Any manipulation of the helper data determining the metric has a 50% chance to insert an error. This does not leak any secret information.

We conclude from the provided attacker model and the discussion in Section II that, to avoid expensive challenge randomization schemes [9], protection of the Loop PUF against SCA needs to hinder the attacker to measure Δ_i or $|\Delta_i|$ in the first place.

IV. THE INTERLEAVED CHALLENGE LOOP PUF

This section introduces the Interleaved Challenge Loop PUF (ICLooPUF), a SCA-hardened enhancement of the Loop PUF described in Section II-A. The fundamental difference is that instead of applying the challenges C and $\neg C$ sequentially, they are applied in an interleaved manner. This is, within a single measurement run, both challenges are alternately applied. When using challenge interleaving, a different measurement concept must be applied, too. The resulting ICLooPUF is shown in Fig. 4a and explained in the following sections.

A. Challenge Interleaving

The Loop PUF consists of an oscillator formed by configurable delay stages and an inverting feedback. When the enable signal is triggered, a rising edge propagates through

[3]Please note, that the same problem is expected to appear if an attacker mounts a side-channel attack on the implementation in [16] and is able to resolve individual RO frequencies.

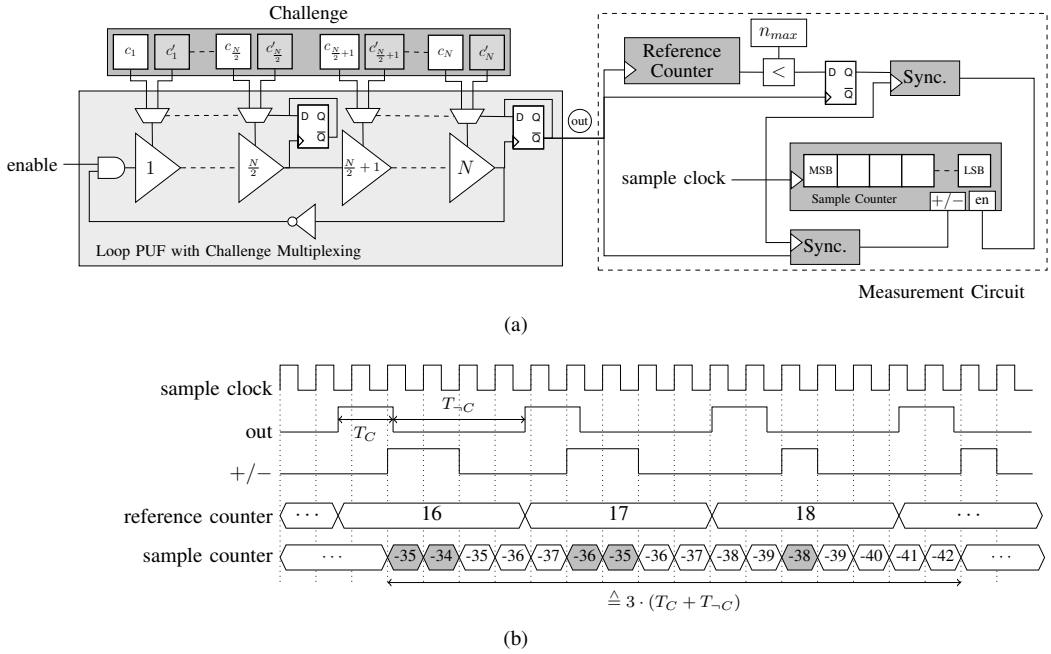


Fig. 4. ICLooPUF: (a) Schematic of challenge interleaving and time-domain measurements, (b) timing diagram of the sample counter.

the delay chain starting from the AND gate in Fig. 4a. Any node in the delay chain passed by the rising edge is set to logical 1 until the falling edge arrives. Assume switching of the challenge bit, which defines the path through a delay element, as soon as the rising edge has passed input and output node of a delay element. Under this assumption, the wave traversing through the ring passes for one period the delay elements defined by challenge $C = [c_0, ..., c_N]$, and for one period the delay elements defined by challenge $\neg C = [c'_0, ..., c'_N]$. The challenge bits are switched by Toggle Flip-Flopss (T-FFs) triggering on the rising edge traversing through the ring. The T-FFs control the multiplexing of the two challenge bits to be interleaved per delay stage.

As shown in Fig. 4a, there is no need to switch every delay stage individually. In particular, it is possible without side-effects to switch any delay element with input and output stabilized to a fixed value. However, since the propagation of the signal is an analog process, the voltage of several nodes can be on an intermediate voltage levels at the same time. Switching such nodes can cause glitches resulting in additional rising edges and effectively errors in the derived secret. Therefore, the number of T-FFs must not be selected too low. We discuss a suitable number of T-FFs for our proof-of-concept design in Section V-A.

B. Time-Domain Measurement

In the original Loop PUF construction from Section II-A a measurement counter counts the oscillations under a specific challenge, and a reference counter stops the counting after a predefined time $T_{ref} \cdot n_{max}$. According to Eq. (1) the value of the measurement counter n_{Loop} is proportional to the Loop PUF frequency.

In the ICLooPUF, we interleave challenges C and $\neg C$, i.e., two alternating periods of length T_C and $T_{\neg C}$ are present at the output. Without challenge interleaving, the expected frequencies of the oscillation are $f_C = \frac{1}{T_C}$ and $f_{\neg C} = \frac{1}{T_{\neg C}}$ for C and $\neg C$. Using the measurement concept of the original Loop PUF would result in a counter value related to the average frequency $\bar{f}_{C,\neg C}$ through

$$\bar{T}_{C,\neg C} = \frac{T_C + T_{\neg C}}{2} \Rightarrow \bar{f}_{C,\neg C} = \frac{2 \cdot f_C \cdot f_{\neg C}}{f_C + f_{\neg C}}. \quad (2)$$

However, the goal of the ICLooPUF is not to measure the average frequency but to directly get the difference $T_C - T_{\neg C}$ between the period lengths under challenges C and $\neg C$. Therefore, different from the original approach, we measure the difference of the two oscillations in the time domain.

For this purpose, in the measurement circuit in Fig. 4a a sample clock with frequency $f_s \gg \max\{f_C, f_{\neg C}\}$ is applied. A counter counts with frequency f_s in an alternating manner upwards and downwards for each one period of the oscillation of the ICLooPUF. For a period length T, the number of oscillations of the sampling clock in this period is $T \cdot f_s$, i.e., the counter value of the sampling counter is related to the period length of the ICLooPUF.

Conditioning the up and down count of the sample counter on the currently applied challenge and with challenge interleaving applied, the counter counts up for each T_C and down for each $T_{\neg C}$. Consequently the counter value after any even number of oscillations of the Loop PUF is related to the difference $T_C - T_{\neg C}$. The higher the sample clock f_s, the more precise the resolution of $T_C - T_{\neg C}$.

The up and down counting in the measurement circuit as well as the enabling of the counter is controlled by the ICLooPUF. The T-FF at the ICLooPUF's output ensures that the output signal is logically 0 for exactly one period and logically 1 for the next period. This corresponds to the control signal for up and down counting. In addition, the ICLooPUF drives a reference counter. The system is stopped, when the

reference counter reaches a pre-defined value n_{max}. This way, for all pairs of challenges and complementary challenges the same number of periods under C and $\neg C$ is used to derive the delay difference. This approach is easy to implement and sample counter values are directly related to the difference in the ICLooPUF's period length for C and $\neg C$.

The transition from the clock domain of the Loop PUF oscillation to the clock domain of the sample clock can result in meta-stability. Therefore, the enable signal of the sample counter as well as the control signal for up and down counting are synchronized with the sample clock. Finally, the comparison result of the reference counter with n_{max} is buffered to prevent wrong results in the sample clock domain.

The timing of the sample counter in relation to the ICLooPUF output signal *out* is depicted in Fig. 4b for some exemplary ICLooPUF periods. The reference counter is increased every $T_C + T_{\neg C}$. The up/down signal (+/-) follows synchronized to the sampling clock with the delay of a two-stage synchronizer. The sample counter counts up (highlighted in gray) when the up/down signal is high and down otherwise. Since $T_{\neg C} > T_C$ in the example, the counter value is overall decreasing in the depicted time interval.

C. Quantization Error

In principle, a quantization error can occur in the suggested measurement circuit. Assuming jitter-free period lengths T_C and $T_{\neg C}$ and $T_s = 1/f_s$ for the sampling clock, and the first rising edge of the sample clock occurring aligned with the ICLooPUF's first rising edge (but not triggering any action), the sampling counter value after n_{max} periods of the output signal generated by the ICLooPUF is

$$\sum_{k=0}^{n_{max}-1} \lfloor [T_C + (k \cdot (T_C + T_{\neg C}) \mod T_s)] \cdot f_s \rfloor - \lfloor [T_{\neg C} + (((k+1) \cdot T_C + k \cdot T_{\neg C}) \mod T_s)] \cdot f_s \rfloor . \quad (3)$$

This equation is construed as: The up and down counting is periodic with $T_C + T_{\neg C}$; in each of these counting periods C and $\neg C$ contribute to the number of increments and decrements with $T_C \cdot f_s$ and $T_{\neg C} \cdot f_s$. However, since T_C and $T_{\neg C}$ are typically not multiples of T_s, a small amount of time expressed by the modulo terms is sampled as part of the wrong period. The quantization error is the difference of Eq. (3) and the expected outcome when n_{max} is reached

$$n_{max}(T_C - T_{\neg C}) \cdot f_s. \quad (4)$$

Although the quantization error can get large in theory,[4] we did not observe such a case in practical measurements. Fig. 5 shows the quantization error, i.e., the difference of Eq. (3) and Eq. (4) for a ICLooPUF frequency in the range from 16 MHz to 16.5 MHz and a sampling clock of 400 MHz. The first experiment in Fig. 5a shows the result without noise, where quantization errors with absolute values of above 10^4 occur. The domains with such a large error are restricted to particular combinations of period lengths T_C and $T_{\neg C}$. Therefore, already a small jitter in the clock frequency significantly reduces the quantization error. This is shown in Fig. 5b, where for each clock cycle a white Gaussian jitter with standard deviation of $\sigma = 100\,\text{ps}$ is simulated. The quantization error is with utmost 1,000 in the same order of amplitude as the variation of the counter values between several measurements we observe in Section V. We therefore conclude that although a quantization error must be considered in theory, it is negligible for practical implementations.

[4]Both terms in Eq. (3) deviate from the best possible quantization value by utmost 1 for each k, so the quantization error is trivially bound by $2 \cdot n_{max}$.

D. Theoretical Analysis of Side-Channel Attack Vectors

The ICLooPUF hides the value of the sample counter, corresponding to $T_C - T_{\neg C}$, from an attacker. This section provides a theoretical analysis of potential side-channel observations and concludes about the robustness of the protection principal. In the following, we consider jitter-free oscillations as the best case from an attacker's perspective.

1) Observation of Measurement Time: First, we consider a possible timing side-channel of the ICLooPUF from observation of the measurement runtime. The attacker observes two alternating oscillations with unknown period lengths T_C and $T_{\neg C}$. Further, the number of oscillations n_{max} is not considered a secret, i.e., it is known by the attacker. Consequently, the measurement for a challenge pair C and $\neg C$ takes $n_{max} \cdot (T_C + T_{\neg C})$ and the attacker can observe the sum $T_C + T_{\neg C}$ of the periods.

2) Observation of Oscillation Frequency: Second, we investigate the spectral side-channel of the ICLooPUF since for oscillation-based PUF primitives, the oscillation frequency is the most important attack vector. We model the oscillation of two interleaved challenges C and $\neg C$ in the time domain as alternating sine waves without jitter and with period lengths of T_C and $T_{\neg C}$ as

$$g(t) = g_1(t) + g_2(t) \quad (5)$$

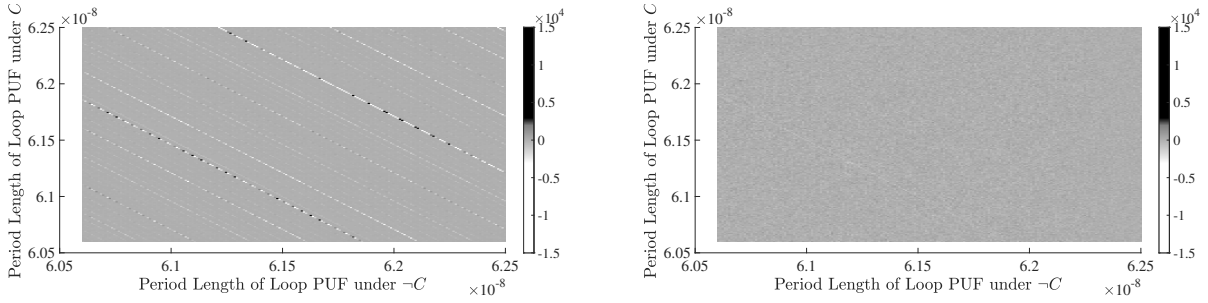
with

$$g_1(t) = \sum_{k=1}^{N} \sin\left(\frac{2\pi}{T_C}\left(t - (k-1)\left(T_C + T_{\neg C}\right)\right)\right) \cdot \Theta\left(\frac{\left(t - \left((k-1)\left(T_C + T_{\neg C}\right) + \frac{T_C}{2}\right)\right)}{T_C}\right) \quad (6)$$

$$g_2(t) = \sum_{k=1}^{N-1} \sin\left(\frac{2\pi}{T_{\neg C}}\left(t - (kT_C + (k-1)\,T_{\neg C})\right)\right) \cdot \Theta\left(\frac{\left(t - \left(k\left(T_C + T_{\neg C}\right)\right) + \frac{T_{\neg C}}{2}\right)}{T_{\neg C}}\right), \quad (7)$$

where $\Theta(t)$ is the rectangular function. For the signal modeled by Eqs. (5) to (7) at each point in time only one of the two oscillations is active, i.e., always one complete period of the oscillation under C is alternated with one complete period under $\neg C$.

Following previous attacks on oscillator-based PUFs, the spectral amplitude is the attack vector for targeting the frequency. Transforming the time domain signal from Eq. (5) to the frequency domain (for details c.f. Appendix A) yields



(a) (b)

Fig. 5. Simulated quantization error (a) without noise, (b) jitter of $\sigma = 100\,\text{ps}$. The period length T of the virtual Loop PUF under the two challenges is sweeped from $T = 60.6\,\text{ns}$ (16.5 MHz) to $T = 62.5\,\text{ns}$ (16 MHz) in steps of 10 ps. The maximum number of iterations is $n_{max} = 2^{18}$; the sampling frequency was set to 400 MHz.

$$|S(f)| = \left| \frac{(1+\gamma)^2 T_C n_{max}}{2\pi} \sum_{n=-N}^{N} \left[\frac{1}{n^2 - (1+\gamma)^2} \left(e^{j2\pi \frac{n}{(1+\gamma)}} - 1 \right) + \frac{\gamma}{n^2\gamma^2 - (1+\gamma)^2} \left(1 - e^{j2\pi \frac{n}{(1+\gamma)}} \right) \right] \cdot \operatorname{sinc}\left(n_{max}(1+\gamma)T_C \left(f - \frac{n}{(1+\gamma)T_C} \right) \right) \right| , \quad (8)$$

where $\gamma := T_{\neg C}/T_C,\ \gamma > 0$ is the ratio of period lengths. In Eq. (8), the global maxima of the sinc function are at multiples of the frequency defined by the sum of the periods $(1+\gamma)T_C = T_C + T_{\neg C}$. The width and the amplitude of the sinc functions are proportional to the sum $T_C + T_{\neg C}$ and the number of oscillations n_{max}. In other words – similar to observations of the measurement time – an attacker can observe the sum of the periods from the frequency spectrum.

3) Conclusion: From the measurement time and the frequency it is not possible to retrieve the secret $T_C - T_{\neg C}$ directly. For both attack vectors the sum $T_C + T_{\neg C}$ of the period lengths can be observed. However, under the assumption from Section III that single periods of T_C and $T_{\neg C}$ cannot be resolved, the sum does not reveal information about its terms, nor about about their difference. Thus, the measurement time and the oscillation frequency of the ICLooPUF are no exploitable attack vectors.

V. Proof-of-Concept Evaluation and Side-Channel Analysis of Challenge Interleaving

In this section, we first provide implementation details of our proof-of-concept design of the ICLooPUF in Section V-A. Second, we provide an evaluation regarding functionality and common PUF metrics in Section V-B. Third, in Section V-C we conduct a side-channel analysis of the ICLooPUF that practically verifies the hardening of the primitive.

A. Practical Implementation

We implement the ICLooPUF on a CW305 board that features an Artix-7 (XC7A100TFTG256) using a sample clock of f_s =400 MHz. In accordance with prior work [7], the design takes a 64-bit challenge, i.e., it consists of 64 delay stages. Since challenges are Hadamard codewords like for the original Loop PUF and the all-zero and all-one challenges are dropped, this results in 63 bits derived from the PUF primitive.

1) Resource Utilization and Place-and-Route: The delay stages of the Loop PUF are realized as Look-Up Tables (LUTs), similar to [7]. We use fixed placement only for LUTs implementing delay stages and challenge interleaving as well as for the T-FFs used for challenge interleaving. In addition, input pins of the mentioned LUTs are fixed. The output T-FF of the Loop PUF is fixed routed to ensure a short feedback. Apart from this, no fixed placement or routing is required in the circuit. In particular, routing between delay stages is left unconstrained since all connections between delay elements are part of the ring for every challenge and cancel out when comparing two frequencies. Hence, only the paths within delay elements can cause a bias that decreases the PUF quality. As for the original Loop PUF from Section II-A the use of Hadamard challenges ensures that bias related to path imbalance is compensated. Finally, we group four delay elements in a single slice for optimum resource usage.[5]

We implement the challenge interleaving by 4-to-2 multiplexers realised in six-input-two-output LUTs. These select from a pair of challenge bits c_i, c_j and the respective complements $\neg c_i, \neg c_j$ either the two challenge bits or the two complementary challenge bits. We implement always two multiplexer-LUTs together with one T-FF triggering the switching of the challenges – implemented by a LUT and a FF – in one slice. This way a Configurable Logic Block (CLB)[6] consists of one slice deriving four challenge bits and one slice with the corresponding four delay elements; The signal from the delay path to the clock input of the T-FF is decoupled through a latch in the slice with the delay elements. The regularity of our design does not only help to reduce bias in the PUF response – since always the same slice in a CLB is used for delay elements – but the existence of macro-blocks also supports a quick and easy design. With these design choices, the number of slices required for the ICLooPUF is twice the amount of slices needed for the original Loop PUF; area optimization at the cost of a more complicated placement and routing is possible.

[5]Internal differences of the LUTs due to their position in a slice and on the FPGA might introduce slight bias. However, these effects are negligible for the design built to analyze side-channel protection.

[6]On Artix-7 FPGAs a CLB consists of two slices, where each slice features four LUTs and eight flip-flops.

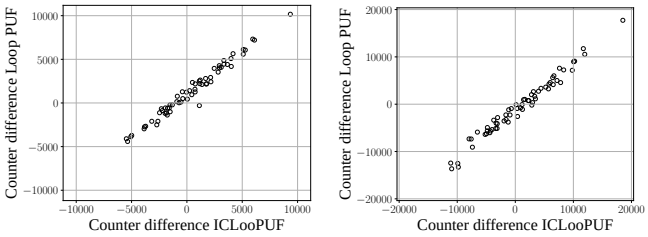
2) Stability Considerations of Interleaving: In order to flip a challenge bit without inserting glitches while the PUF is oscillating, the same stable state must be applied at input and output of the delay elements. In other words, the delay of the feedback path from an inner node of the delay chain, through the T-FF to the multiplexer switching the challenge in addition to the delay from the challenge multiplexer input back to the delay element must be smaller than a half-period of the oscillation.

For our design the oscillation frequency is around $16\,\mathrm{MHz}$ corresponding to a half-period of $31.25\,\mathrm{ns}$. Without dedicated place-and-route of the feedback, switching tuples of four delay stages of the 64 stages provides a sufficiently low delay. Increasing the number of stages that switch challenge bits in parallel would likely be possible, in particular if manual place-and-route would be applied for the feedback path. This would reduce resource allocation and increase the design effort but is considered out-of-scope for this work.

3) Offset Compensation: The oscillation of the ICLooPUF has a lower slew-rate compared to a normal clock signal. Combined with the specific propagation delay of the FPGA's internal gates, we observed that one (e.g., the rising) edge propagates faster than the other (e.g., the falling) edge from the output T-FF of the ICLooPUF to the sample counter's input that selects up or down count. Consequently, the count of one half period of the oscillation of the ICLooPUF is extended compared to the other half period, causing an offset of the sample counter value.

To compensate the device-specific offset, we extend the measurement of the ICLooPUF to two phases: First, we apply C and $\neg C$ as interleaved challenges for ${}^{n_{max}}/_{2}$ periods. Neglecting noise, the sample counter value is $n_s^+ = n_d + n_o$, where n_d is the actual difference and n_o is the offset of the counter due to the asymmetric delay of the connect from the delay chain to the sample counter. Second, we exchange the challenges and apply $\neg C$ as first challenge and C as second challenge for another ${}^{n_{max}}/_{2}$ oscillation periods, such that the sample counter value is $n_s^- = -n_d + n_o$. The delay difference of oscillations under C and $\neg C$ is computed from the difference $n_s^+ - n_s^- = 2 \cdot n_d$ on the device. Since the delay offset n_o is independent of the order of the challenges and constant for the same device, it cancels out in the difference.

4) Extensions for Experiments: For the experiments we add some extensions to our design. First, we are able to send arbitrary values of challenges C, $\neg C$ to the device for interleaving; in an actual design, challenges C could be generated on-chip and $\neg C$ would be generated by inverting C on-chip. Second, we implement different modes for our design: An *interleaved* mode implements the challenge interleaving from Section IV-A; Two challenges are interleaved and the counter is counting down when the first challenge C is applied and up for the second challenge $\neg C$. A *sequential* mode applies a single challenge C or $\neg C$ without interleaving, i.e., as for the original Loop PUF. The *sequential* mode is leveraged to verify the functionality of the *interleaved* mode in Section V-B and to determine the expected oscillation frequency of the ICLooPUF for SCA experiments in Section V-C. Third, we can set for our experiments the reference value n_{max}, and the design allows for reading out the value n_{loop} of the counter for PUF evaluation and as reference for SCA.



(a) (b)

Fig. 6. Comparison of counter value n_{loop} averaged over 1000 repetitions for ICLooPUF and counter value difference achieved without challenge interleaving (original Loop PUF) applying each the same challenges $\neg C$ and C. (a) $n_{max} = 2^{17}$, (b) $n_{max} = 2^{18}$.

B. Evaluation of the ICLooPUF

In this section we evaluate the challenge interleaving of the ICLooPUF compared to sequential measurements of the original Loop PUF. Additionally, we provide a *preliminary* PUF quality assessment, which shows possible trade-offs between reliability and runtime.

1) Equivalence of Interleaved and Sequential Mode: Fig. 6 shows the average counter values for 1000 repetitions per challenge of the *interleaved* mode compared to the difference of averaged counter values for $\neg C$ and C in *sequential* mode, which is equivalent to the sequential operation of the original Loop PUF. The plots show a linear relationship, i.e., the two modes lead to the same results. Furthermore, doubling the reference counter values n_{max} yields doubled counter values for both operation modes, which is the expected behavior due to the doubled measurement time. We conclude that the ICLooPUF is functionally equivalent to the original Loop PUF.

2) Preliminary PUF Quality Assessment: In order to provide a first assessment[7] of the quality of the ICLooPUF, we evaluate the sign-based bit derivation method from Section II-B and the TMHD from Section II-C. As the TMHD makes use of helper data, we also provide results for the sign-based method and *dark-bit masking*, i.e., instead of all 63 bits only the l most reliable bits are used. Neglecting specialized encoding schemes, storing the reliability information requires 63 bits of helper data as for TMHD, which enables a fair comparison of both derivation methods. Table II provides results regarding the common PUF metrics *reliability*, *uniqueness*, and *uniformity* – computed as in [24] – for varying values of the reference counter value n_{max}. As expected, uniformity and uniqueness are independent of the reference counter value n_{max}.[8] Both metrics are close to their optimal values of 0.5.

Regarding reliability, Fig. 7 depicts the values from Table II for an easier comparison. The results provide further motivation to use the TMHD approach: for $n_{max} \geq 2^{17}$ the method leads to nearly perfectly reliable reconstruction under

[7] An in-depth evaluation requires significantly more data and devices. As we focus on SCA hardening of the primitive, this is considered future work.

[8] The number of bits l does not have an effect either, therefore we omit the respective rows in Table II as they do not carry information.

TABLE II
PUF METRICS FOR 10 DEVICES, l BITS, AND 100 REPETITIONS (FROM WHICH 10 ARE TAKEN FOR ENROLLMENT).

			n_{max}								
		l	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}
Reliability	Sign	63	0.797	0.834	0.880	0.909	0.935	0.956	0.969	0.978	0.983
	Sign	60	0.809	0.850	0.895	0.925	0.953	0.972	0.984	0.991	0.996
	Sign	50	0.850	0.893	0.935	0.965	0.982	0.992	0.998	1.000	1.000
	Sign	32	0.912	0.951	0.979	0.991	0.997	1.000	1.000	1.000	1.000
	TMHD	63	0.782	0.846	0.910	0.952	0.978	0.993	0.998	1.000	1.000
Uniformity	Sign	63	0.508	0.510	0.514	0.522	0.525	0.529	0.525	0.525	0.525
	TMHD	63	0.479	0.479	0.475	0.471	0.475	0.470	0.476	0.481	0.470
Uniqueness	Sign	63	0.482	0.481	0.485	0.480	0.482	0.474	0.472	0.479	0.476
	TMHD	63	0.487	0.499	0.504	0.494	0.500	0.490	0.493	0.493	0.486

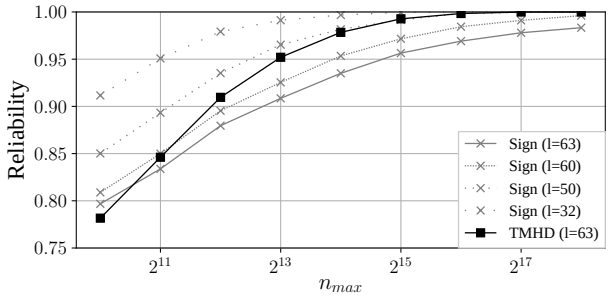


Fig. 7. Reliability for 10 devices, and 100 repetitions; 10 of the repetitions are taken to derive the expected responses (enrollment).

nominal conditions. On the other hand, using the sign-based method a similar reliability can be achieved for taking only the $l = 50$ most reliable bits, i.e., deriving 13 bits less than for the TMHD. Note that the runtime to derive the PUF response is proportional to n_{max}, i.e., the TMHD provides the best trade-off regarding runtime and derived bits, whereas for the sign-based derivation either a significant loss of bits has to be tolerated or the runtime has to be increased to guarantee a stable response.

Summing up, the proof-of-concept design of the ICLooPUF is a highly reliable PUF primitive. The use of the TMHD improves the number extracted bits for a targeted reliability and runtime compared to the sign-based method.

C. Experimental Side-Channel Evaluation

Following previous work targeting the Loop PUF [7], [9], we measure the power side-channel over the CW305 board's shunt resistor. The time domain measurements acquired by a PicoScope 6402 USB-oscilloscope at sampling frequency f_s =156.25 MS/s are transformed into the frequency domain. In accordance with the results from Section V-B the reference counter values is set to a large value of $n_{max} = 2^{18}$; this results in high reliability and is a best case for the attacker who can accumulate more information.

1) Exploration of Frequencies of Interest: The original Loop PUF leaks by the oscillation frequency. Considering in addition Eq. (8), the dependency between the frequency difference of the PUF with C and $\neg C$ applied on the one hand and the amplitude, width and frequency of the ICLooPUF's spectrum on the other hand has to be investigated. In order to enable the analysis, the *sequential* mode described in Section V-A allows to determine the Frequency of Interest (FoI) range. For each challenge, we acquire the counter value n_{loop} and calculate the frequency of the delay chain as

$$f_{oI} = \frac{2 \cdot n_{max}}{n_{loop} \cdot T_{ref}}. \tag{9}$$

Note that due to the output T-FF, which acts as a frequency divider, the reference counter measures only half of the frequency and the factor of 2 has to be added to determine the oscillation frequency of the loop. The average frequency across all challenges C_i and $\neg C_i$, $i \in \{1, \ldots, 63\}$ is $\bar{f}_{oI} = 16.021\,\mathrm{MHz}$ with a frequency range $16.007\,\mathrm{MHz} \leq f_{oI} \leq 16.035\,\mathrm{MHz}$, which defines the region of interest for the following SCA evaluation.

2) Side-Channel Results: Considering the average frequency of $\bar{f}_{oI} = 16.021\,\mathrm{MHz}$ and the value $n_{max} = 2^{18}$, the expected runtime of the ICLooPUF per challenge is around $32.7\,\mathrm{ms}$, from which the first $30\,\mathrm{ms}$ are transformed into the frequency domain.[9] Finally, for easier peak detection, the frequency spectrum is low-pass filtered.

Fig. 8 shows exemplary spectra in the FoI range from $16\,\mathrm{MHz}$ to $16.05\,\mathrm{MHz}$. Figs. 8a and 8b correspond to challenges C and $\neg C$ with the maximum and minimum counter difference in *sequential* mode, i.e., the extreme values an attacker can observe for the original Loop PUF. In Fig. 8c the spectra for the same challenges in *interleaved* mode are depicted, where the solid line corresponds to Fig. 8a, and the dashed line corresponds to Fig. 8b. Note that the increased amplitude in Fig. 8c compared to Figs. 8a and 8b stems from the fact that only in *interleaved* mode the challenges are switched by T-FFs, while in *sequential* mode the T-FFs are deactivated, i.e., the increased amplitude is caused by the additional switching activity. The difference of frequencies in the spectra in Figs. 8a and 8b corresponds to the underlying counter differences, i.e., an attacker can learn from the peak comparison. On the other hand, in Fig. 8c the depicted extreme cases show similar spectra corresponding to the *averaged* frequency of the ICLooPUF challenged with C and $\neg C$ interleaved. At first glance, there is no obvious attack vector

[9] Additionally, we determine a noise floor by connecting the system clock instead of the delay chain to the counting circuitry. Subtracting the noise floor from the actual signal allows to remove regular components of the Device Under Test (DUT), such as the clock frequency, for better detecting relevant frequencies, but is not a necessary condition for the attack.

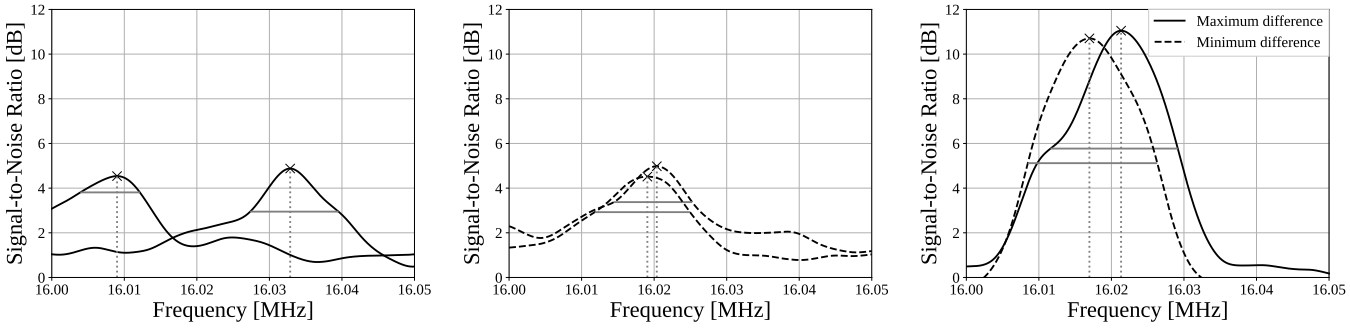


Fig. 8. Frequency spectra revealed from side-channel analysis in the FoI range for extreme cases of expected frequencies. Challenges C_i and $\neg C_i$ with (a) maximum counter difference and (b) minimum counter difference in *sequential* mode, and (c) interleaved challenges corresponding to (a) and (b).

TABLE III
SIDE-CHANNEL ANALYSIS RESULTS: CORRELATION OF COUNTER VALUES AND PHYSICAL OBSERVATIONS AVERAGED OVER DIFFERENT NUMBERS OF MEASUREMENT TRACES AND FOR $n_{max} = 2^{18}$ AND $N = 63$.

	Traces	Frequency	Amplitude	Width
Loop PUF (C)	1	-0.961	0.122	-0.115
Loop PUF (C)	10	-0.995	-0.213	-0.309
Loop PUF ($\neg C$)	1	-0.970	-0.103	0.089
Loop PUF ($\neg C$)	10	-0.987	0.018	-0.109
ICLooPUF (Δ_i)	1	0.173	-0.135	-0.003
ICLooPUF (Δ_i)	10	0.156	-0.073	-0.028
ICLooPUF (Δ_i)	100	0.233	-0.104	-0.067
ICLooPUF ($\lvert\Delta_i\rvert$)	1	-0.038	0.041	0.202
ICLooPUF ($\lvert\Delta_i\rvert$)	10	0.098	0.093	0.409
ICLooPUF ($\lvert\Delta_i\rvert$)	100	0.181	0.128	0.406

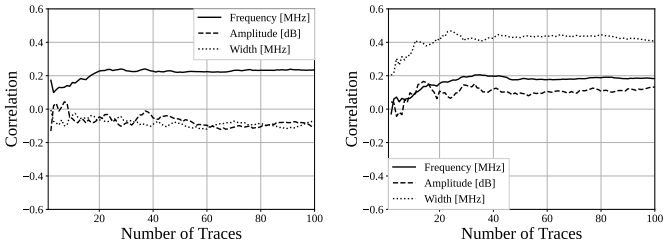


Fig. 9. Evolution of correlation between averaged counter values and observed side-channel properties for 63 challenges of the ICLooPUF, same device as used for Table III. (a) correlation with the counter values Δ, (b) correlation with the absolute counter values $|\Delta|$.

on the ICLooPUF's spectrum visible, so we investigate further.

In order to further evaluate the side-channel resistance of the ICLooPUF, we compare the real counter values and their amplitudes with different properties of the observed frequency peaks. As mentioned above, we consider the amplitude, the frequency and the width of the peak, which are marked in Fig. 8 as cross, dotted gray vertical line and solid gray horizontal line respectively. Pearson's correlation coefficient of the counter values with the characteristics of the peak is used as a measure for similarity, i.e., the predictability of the counter values for an SCA adversary. In Table III the correlations between peak characteristics and counter values are provided for the original Loop PUF as well as for the ICLooPUF. The lower part of the table provides correlations of the absolute counter values to analyse the impact on bit derivation with amplitude-based approaches. An attacker could try to aggregate information from several observations by using repeated measurements of the same challenge. Therefore, in Table III the peak characteristics are determined from the average of several measurement traces of the same challenge and compared to *averaged counter values*.

Even with a single measurement per challenge in *sequential* mode the match between the frequency and the counter value leads to an absolute correlation of above 0.961,[10] i.e., as expected there is a direct relationship between frequency and counter value for the original Loop PUF. Adding further measurements increases the correlation. For the original Loop PUF, the observed frequencies of C and $\neg C$ therefore reveal sign and amplitude of the frequency difference, and hence the secret [7], [9].

In Table III, frequency, amplitude, and width show correlations of up to 0.409 with the counter values and the absolute counter value $|\Delta_i|$ for the ICLooPUF.[11] A detailed insight of the evolution of the correlation is provided in Fig. 9, where Figs. 9a and 9b depict the the correlation with increasing number of averaged measurement traces per challenge for the counter values respectively their absolute value. In Fig. 9a, the correlations are below 0.25. Similarly, in Fig. 9b correlations with frequency and amplitude converge towards values of below 0.2 with increasing measurements, and the width to around 0.4. In other words increasing the number of repetitions beyond the depicted number does not improve the match of observation and counter values for the ICLooPUF.

Even though correlation values of 0.4 do not indicate a causal relation with the counter values, we investigate whether the same correlation is observed on a different device to rule out any profiling attacks. Fig. 10 depicts the correlation on a second device: the maximum absolute correlation is 0.31,[12] but the sign differs compared to Fig. 9. Therefore, even if

[10]For $N = 63$ observations, the confidence interval of the correlation coefficient is within $[0.9361, 0.9763]$ with a confidence level of 0.95.

[11]The confidence interval of the correlation coefficient is within $[0.1794, 0.5963]$ with a confidence level of 0.95.

[12]The confidence interval of the correlation coefficient is within $[0.0674, 0.5180]$ with a confidence level of 0.95.

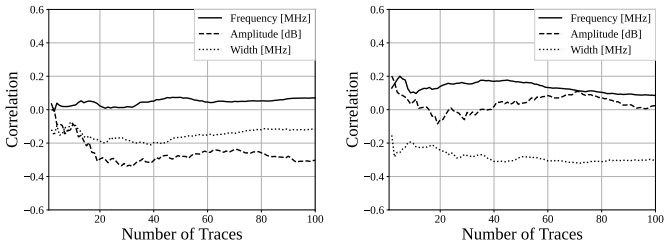


Fig. 10. Evolution of correlation between averaged counter values and observed side-channel properties for 63 challenges of the ICLooPUF, different device as used for Table III. (a) correlation with the counter values Δ, (b) correlation with the amplitude $|\Delta|$ of the counter value.

correlations would theoretically allow for reducing entropy on a particular device – which is not indicated by the measured correlations – an attacker could not derive the device-specific correlation from profiling on a second board.

Finally note that, e.g., for an attack on the TMHD scheme, the intervals for bit derivation would be wrongly estimated if there is no deterministic relation between observations and counter values, i.e., the required precision for an attack is not given by weak correlations.

Summing up, we conclude that the ICLooPUF does not leak exploitable side-channel information in the frequency domain via frequency, amplitude or width of peak in the FoI range. Therefore, it constitutes a SCA-hardened PUF primitive.

VI. Discussion

The ICLooPUF suggested in this work provides resilience against side-channel attacks in combination with sign-based as well as with amplitude-based bit derivation such as the TMHD scheme. It is compared against state-of-the-art countermeasures in Table IV.

Regarding the hardware overhead of our N-bit design, it requires N additional 2-bit multiplexers – which can be replaced by XOR gates inverting C for every second period – and utmost N T-FFs for switching challenge bits; in our proof-of-concept design we used $N/4$ T-FFs. Different from the Loop PUF, the ICLooPUF uses time-domain sampling with an up/down counter for measurement. Although the counters and the comparator are arranged differently, the only overhead for the measurement circuit is the synchronization, which is negligible. The protection mechanism has therefore overall low hardware overhead when compared to the original Loop PUF. It is also smaller than the challenge randomization (CR) approach [9], which needs to implement a randomization algorithm and a TRNG. Compared to temporal masking (TM) [5] the synchronization and T-FFs for challenge switching constitute some overhead; TM requires $N+1$ XOR gates and a 1-bit storage on top of the original Loop PUF.

The ICLooPUF has no timing overhead compared to the original Loop PUF and temporal masking. The design is, however, faster than challenge randomization, which has a probabilistic runtime due to possible re-sampling of duplicate indices, and requires additional time for randomizing the challenge order. It is worth mentioning that – different from the state of the art – our new design does not require any random bits.

TABLE IV
Comparison of countermeasures for the Loop PUF.
TM=Temporal Masking, CR=Challenge Randomization.

	TM [7]	CR [9]	ICLooPUF
Hardware Overhead	very low	high	low
Timing	deterministic	probabilistic	deterministic
Randomness source	Loop PUF	TRNG	not needed
Random bits	N	$> N\lceil\log_2(N)\rceil$	0
Sign-based	x	x	x
Amplitude-based	–/(x)	x	x

Finally we stress, that only challenge randomization and the ICLooPUF protect against SCA when using sign-based as well as amplitude-based quantization. Temporal masking can only protect the sign and has limited (in case of equiprobable quantization or order encoding) or no effect (in case of TMHD) as discussed in Section II-D. Overall, if the reduced reliability of sign-based bit derivation can be tolerated at the cost of error correction or reduced entropy, temporal masking has lower resource overhead. However, in order to enable highly efficient and reliable bit derivation, amplitude-based schemes are needed. Our new design maintains a low hardware overhead in the presence of an SCA adversary and prevents the need for a separate TRNG compared to challenge randomization when using these schemes.

In conclusion, the ICLooPUF is an oscillator-based PUF without side-channel leakage particularly suitable to protect amplitude-based bit derivation.

VII. Conclusion

In this work, we introduced a new PUF primitive: the Interleaved Challenge Loop PUF (ICLooPUF). It is based on the Loop PUF and uses challenge interleaving instead of sequential challenges to protect against SCA. The countermeasure applies in particular to bit derivation from amplitude-based scheme such as the TMHD scheme, but can be used with sign-based bit derivation as well. It comes with low hardware overhead and needs no random numbers for protection. We provided a theoretical justification for the protection principle. Further, we demonstrated the practical application of the concept, and conducted a SCA evaluation that practically confirms the protection.

Appendix

A. Derivation of frequency representation

The signal $g(t)$ from Eq. (5) is periodic with $P = T_C + T_{\neg C}$, and can be approximated by a Fourier series as an infinite signal

$$g_N(t) = \sum_{n=-N}^{N} c_n \cdot e^{j\frac{2\pi}{P}nt}, \quad c_n = \frac{1}{P}\int_P g(t)\cdot e^{-j\frac{2\pi}{P}nt}\,dt.$$

$$g_N(t) = \frac{T_C + T_{\neg C}}{2\pi} \sum_{n=-N}^{N} (\alpha(n) - \beta(n))\, e^{j\frac{2\pi n}{T_C+T_{\neg C}}(t-T_C)} + (\beta(n) - \alpha(n))\, e^{j\frac{2\pi}{T_C+T_{\neg C}}nt} \tag{10}$$

with

$$\alpha(n) = \frac{T_C}{n^2 T_C^2 - (T_C + T_{\neg C})^2},\ \beta(n) = \frac{T_{\neg C}}{n^2 T_{\neg C}^2 - (T_C + T_{\neg C})^2}.$$

Transforming Eq. (10) into the frequency domain using the Fourier transform shows that the frequency of the interleaved signal is only present for multiples of the average frequency $f = \frac{n}{T_C + T_{\neg C}}$

$$S(f) = \frac{T_C + T_{\neg C}}{2\pi} \sum_{n=-N}^{N} \left[(\alpha(n) - \beta(n))\, e^{j2\pi f T_C} + (\beta(n) - \alpha(n)) \right] \delta\left(f - \frac{n}{T_C + T_{\neg C}} \right) \tag{11}$$

with decreasing amplitudes for $|n| \to \infty$ as $\alpha(n), \beta(n) \sim \frac{1}{n^2}$. Setting $T_{\neg C} := \gamma T_C,\ \gamma > 0$, i.e., $T_C(1+\gamma) = T_C + T_{\neg C} = P$ the spectral amplitude can be expressed as the relative period of the interleaved signals

$$|S(n,\gamma)| = \frac{(1+\gamma)}{2\pi} \left| \frac{1}{n^2 - (1+\gamma)^2} \left(e^{j2\pi \frac{n}{(1+\gamma)}} - 1 \right) + \frac{\gamma}{n^2\gamma^2 - (1+\gamma)^2} \left(1 - e^{j2\pi \frac{n}{(1+\gamma)}} \right) \right|, \tag{12}$$

i.e., the magnitude of the spectral components depends on the relative period γ.

Finally, the result in Eq. (11) represents the ideal spectrum for an infinite signal $g_N(t)$. However, in reality the signal $g_N(t)$ is limited to n_{max} clock cycles of the period $P = T_C + T_{\neg C}$. Transforming the time limited signal $g_N(t) \cdot \Theta\left(\frac{t}{n_{max} \cdot (T_C + T_{\neg C})} \right)$ using the properties of the Fourier transform $e^{jat} \cdot \Theta(bt) \overset{\mathcal{F}(\cdot)}{\Longleftrightarrow} \delta\left(f - \frac{a}{2\pi}\right) * \frac{1}{|b|} \operatorname{sinc}\left(\frac{f}{b}\right) = \frac{1}{|b|} \operatorname{sinc}\left(\frac{f - \frac{a}{2\pi}}{b}\right)$ yields

$$|S(f)| = \left| \frac{(1+\gamma)^2 T_C n_{max}}{2\pi} \sum_{n=-N}^{N} \left[\frac{1}{n^2 - (1+\gamma)^2} \left(e^{j2\pi \frac{n}{(1+\gamma)}} - 1 \right) + \frac{\gamma}{n^2\gamma^2 - (1+\gamma)^2} \left(1 - e^{j2\pi \frac{n}{(1+\gamma)}} \right) \right] \cdot \operatorname{sinc}\left(n_{max}(1+\gamma) T_C \left(f - \frac{n}{(1+\gamma) T_C} \right) \right) \right|,$$

where $\operatorname{sinc}(f) = \sin(\pi f)/(\pi f)$.

REFERENCES

[1] Z. Cherif, J. Danger, S. Guilley, and L. Bossuet, "An easy-to-design PUF based on a single oscillator: The loop PUF," in *2012 15th Euromicro Conference on Digital System Design*, Sep. 2012, pp. 156–162.

[2] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Semi-invasive EM attack on FPGA RO PUFs and countermeasures," in *6th Workshop on Embedded Systems Security (WESS'2011)*. ACM, Mar 2011.

[3] D. Merli, J. Heyszl, B. Heinz, D. Schuster, F. Stumpf, and G. Sigl, "Localized electromagnetic analysis of RO PUFs," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, June 2013, pp. 19–24.

[4] M. Shiozaki and T. Fujino, "Simple electromagnetic analysis attacks based on geometric leak on an ASIC implementation of ring-oscillator PUF," in *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop*, ser. ASHES'19. New York, NY, USA: ACM, 2019, pp. 13–21.

[5] L. Tebelmann, M. Pehl, and V. Immler, "Side-channel analysis of the TERO PUF," in *Constructive Side-Channel Analysis and Secure Design*, I. Polian and M. Stöttinger, Eds. Springer International Publishing, 2019, pp. 43–60.

[6] U. Mureddu, B. Colombier, N. Bochard, L. Bossuet, and V. Fischer, "Transient effect ring oscillators leak too," in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 2019, pp. 37–42.

[7] L. Tebelmann, J.-L. Danger, and M. Pehl, "Self-secured PUF: Protecting the loop PUF by masking," in *Constructive Side-Channel Analysis and Secure Design*, G. M. Bertoni and F. Regazzoni, Eds. Cham: Springer International Publishing, 2020, pp. 293–314.

[8] J. Danger, S. Guilley, and A. Schaub, "Two-metric helper data for highly robust and secure delay PUFs," in *2019 IEEE 8th International Workshop on Advances in Sensors and Interfaces (IWASI)*, June 2019, pp. 184–188.

[9] L. Tebelmann, U. Kühne, J.-L. Danger, and M. Pehl, "Analysis and protection of the two-metric helper data scheme," in *Constructive Side-Channel Analysis and Secure Design*, S. Bhasin and F. De Santis, Eds. Cham: Springer International Publishing, 2021, pp. 279–302.

[10] O. Rioul, P. Solé, S. Guilley, and J.-L. Danger, "On the entropy of physically unclonable functions," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 2928–2932.

[11] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, ser. CCS '99. ACM, 1999, pp. 28–36.

[12] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 523–540.

[13] M. Pehl, M. Hiller, and G. Sigl, "Secret key generation for physical unclonable functions," in *Information Theoretic Security and Privacy of Information Systems*, R. F. Schaefer, H. Boche, A. Khisti, and H. V. Poor, Eds. Cambridge University Press, Mar 2017, ch. Secret Key Generation and Authentication, pp. 362–389.

[14] T. Stanko, F. Nur Andini, and B. Škorić, "Optimized quantization in zero leakage helper data systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1957–1966, 2017.

[15] V. Immler and K. Uppund, "New insights to key derivation for tamper-evident physical unclonable functions," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, p. 30–65, May 2019.

[16] R. Maes, A. Van Herrewege, and I. Verbauwhede, ""PUFKY: A fully functional PUF-based cryptographic key generator"," in *Cryptographic Hardware and Embedded Systems – CHES 2012*, E. Prouff and P. Schaumont, Eds. Springer Berlin Heidelberg, 2012, pp. 302–319.

[17] D. Karakoyunlu and B. Sunar, "Differential template attacks on PUF enabled cryptographic devices," *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2010.

[18] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Side-channel analysis of PUFs and fuzzy extractors," in *Trust and Trustworthy Computing*, ser. Lecture Notes in Computer Science, J. M. McCune, B. Balacheff, A. Perrig, A.-R. Sadeghi, A. Sasse, and Y. Beres, Eds. Springer Berlin Heidelberg, 2011, no. 6740, pp. 33–47.

[19] D. Merli, F. Stumpf, and G. Sigl, "Protecting PUF error correction by codeword masking," *IACR Cryptology ePrint Archive*, vol. 334, 2013. [Online]. Available: http://eprint.iacr.org/2013/334

[20] L. Tebelmann, M. Pehl, and G. Sigl, "EM side-channel analysis of BCH-based error correction for PUF-based key generation," in *Proceedings of the 2017 Workshop on Attacks and Solutions in Hardware Security*, ser. ASHES '17. New York, NY, USA: ACM, 2017, pp. 43–52.

[21] E. Strieder, C. Frisch, and M. Pehl, "Machine learning of physical unclonable functions using helper data: Revealing a pitfall in the fuzzy commitment scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, no. 2, pp. 1–36, 2021.

[22] M. Hiller, M. Weiner, L. Rodrigues Lima, M. Birkner, and G. Sigl, "Breaking through fixed PUF block limitations with differential sequence coding and convolutional codes," in *Proceedings of the 3rd International Workshop on Trustworthy Embedded Devices*. New York, NY, USA: Association for Computing Machinery, 2013, p. 43–54.

[23] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 34, no. 6, pp. 889–902, 2015.

[24] A. Maiti, V. Gunreddy, and P. Schaumont, *A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions*. New York, NY: Springer New York, 2013, pp. 245–267.